MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2018/2019

DMT5018 – MICROCONTROLLER TECHNOLOGY

(Diploma in Electronic Engineering)

23 OCTOBER 2018
9:00 AM – 11:00 AM
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This question paper consists of 8 pages (including the cover page & appendices).
2. Answer **ALL** questions. All necessary working steps must be shown.
3. Write all your answers in the answer booklet provided.

QUESTION 1 [25 Marks]

a) Compare the differences between microprocessor and microcontroller from the following perspectives.

(i) Hardware Architecture [2 marks]

(ii) Application [2 marks]

(iii) Instruction Set [2 marks]

b) If a 16-bit microprocessor is to be used in a computer system, what is the size of its address bus, data bus and registers if it needs to access a memory with size 512 Mbytes of data (maximum)? Assume that each memory location can store 16-bits of data. [4 marks]

c) Draw the internal block diagram of the 8051 microcontroller. State which are the on-chip peripheral devices. [8 marks]

d) An 8051 assembly language instruction sequence is stored in the memory as shown in the Table Q1 (d). Fill in the sequence of values in the program counter (PC), the address register (AR), the data register (DR) and the accumulator (ACC) if the microcontroller executes the instructions starting with ACC being "0AH".

Table Q1 (d)

Address	Contents	Instruction	PC	AR	DR	ACC
0100H	74H	MOV A, #55H			74H	
0101H	55H		0102H			
0102H	03H	RR A		0102H		
0103H	80H	SJMP 0100H			80H	
0104H	FBH				FBH	AAH

[7 marks]

Continued...

QUESTION 2 [25 Marks]

a) Show the alternative pin functions for the Port 3 of 8051 microcontroller. [4 marks]

b) Illustrate with examples, the differences between MOVC and MOVX instructions. [4 marks]

c) Compute the sequence of status in the carry flag (CY), auxiliary carry flag (AC), overflow flag (OV) and parity flag (P) if the 8051 assembly language instruction sequence shown in Table 2 (c) is executed?

Table 2 (c)

Instruction	CY	AC	OV	P
MOV A, #5FH				
MOV B, #2CH				
ADD A, B				
MUL AB				

[8 marks]

d) An 8051 microcontroller assembly language subroutine sequence is as shown in Table 2 (d)

Table 2 (d)

```
SUB:    MOV R0, #20H
LOOP:   MOV @R0, #0
        INC R0
        CJNE R0, #80H, LOOP
        RET
```

(i) How many machine cycles does each instruction need to execute? [3 marks]

(ii) Convert the subroutine to machine language. [3 marks]

(iii) How long does this subroutine take to execute?
[Hint: Assume a crystal frequency of 12MHz] [3 marks]

Continued...

QUESTION 3 [25 Marks]

(a) Consider an external crystal oscillator with a frequency of 16MHz instead of the standard 12MHz. Write a program to generate a periodic waveform on P1.0 with the highest possible frequency. Hence, calculate the frequency and duty cycle of that waveform.

[5 marks]

(b) A buzzer is connected to P1.7 as shown in the Fig Q3 (b), and a debounced switch is connected to P1.6. Write a program that reads the logic level provided by the switch and sounds the buzzer for 1 second for each 1-to-0 transition detected.

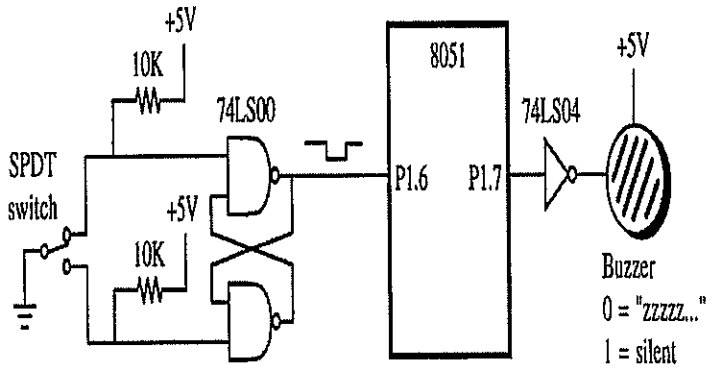


Fig. Q3 (b)

[8 marks]

(c)

(i) Write an instruction sequence to initialize the serial port to operate as a 9-bit UART at 9600 baud. Use Timer1 to provide the baud rate clock, assuming that the crystal oscillator frequency, f_{osc} = 12 MHz.

[6 marks]

(ii) Is the result baud rate exactly 9600 baud? If not, what is the percentage error(%error)?

[6 marks]

Continued...

QUESTION 4 [25 Marks]

(a) Briefly illustrate the term "interrupt vector".

[2 marks]

(b) Show the steps required to use an interrupt in the 8051 family.

[3 marks]

(c) Can RESET be considered as an interrupt? Explain briefly.

[3 marks]

(d) Identify the 8051 microcontroller interrupt flags that must be cleared by software once an interrupt has been accepted.

[2 marks]

(e) Assume that the $\overline{INT0}$ pin is connected to a switch. Whenever it is pressed, it should turn ON an LED that is connected to P1.2. When it is turned on, it should stay on for a fraction of a second. Write a program to perform this task.

[9 marks]

(f) External Interrupt 1, Timer 0 interrupt and the serial port interrupt are to be implemented in an 8051 system. The External Interrupt 1 should have the lowest priority level.

(i) Write a sequence of assembly language instructions to implement the above.

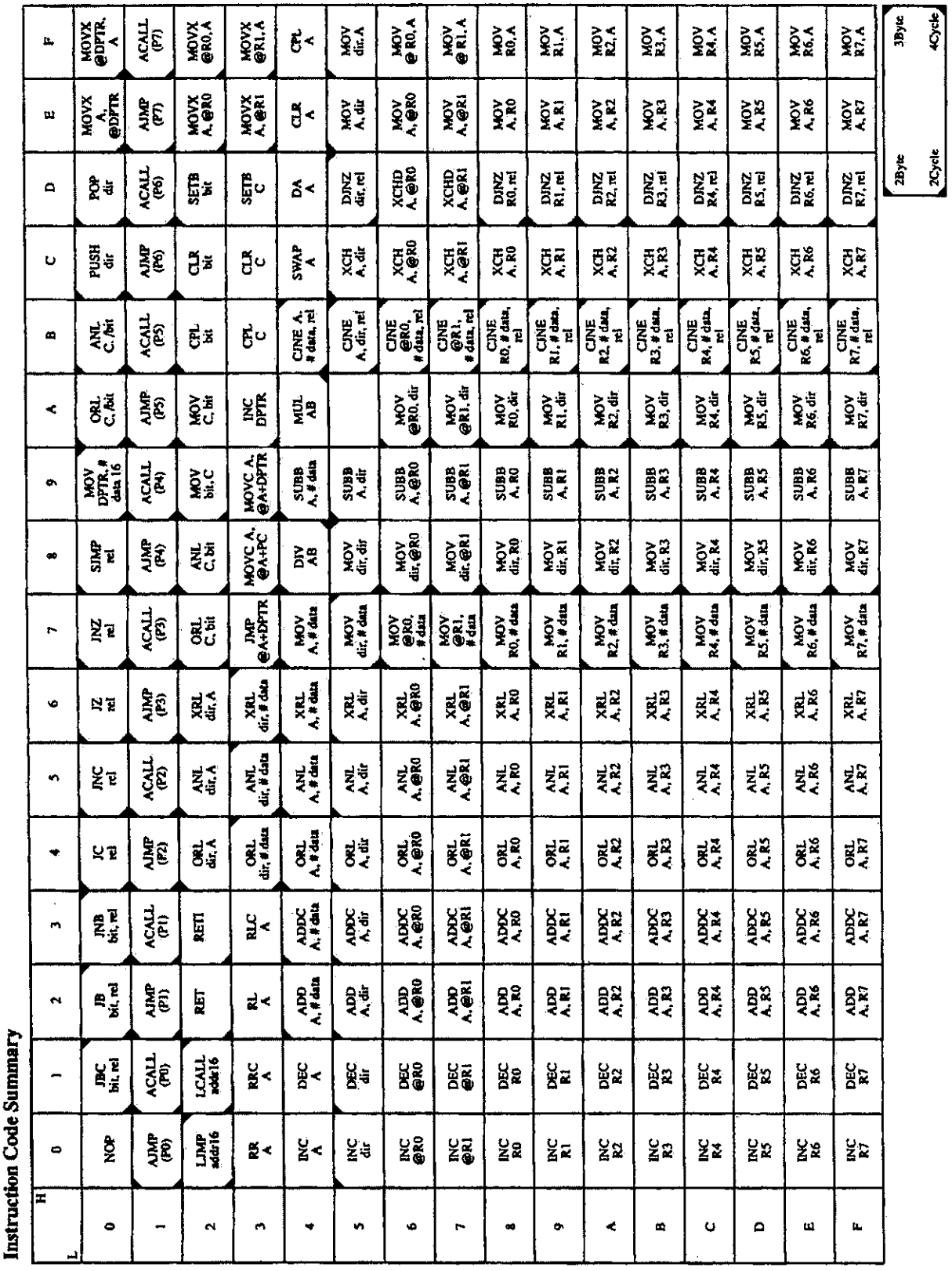
[4 marks]

(ii) What is the sequence of servicing the interrupts if all the mentioned interrupts above occur simultaneously?

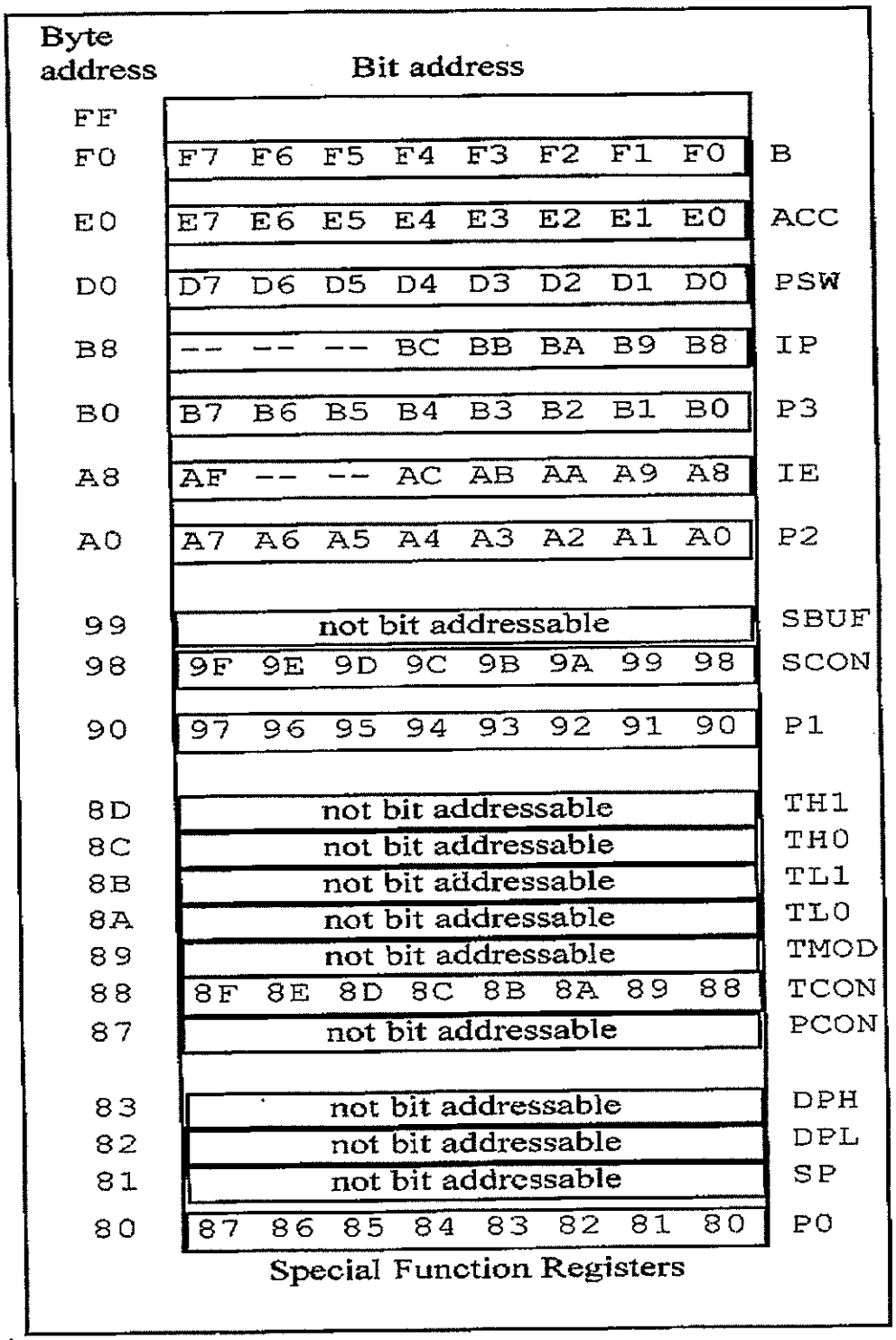
[2 marks]

End of Page.

APPENDIX A: OPCODE MAP


Instruction Code Summary
H
L
0 NOP | JBC bit, rel | JB bit, rel | JNB bit, rel | JC rel | JNC rel | JZ rel | JNZ rel | SJMP rel | MOV DPTR, # data 16 | ORL C, /bit | ANL C, bit | PUSH dir | POP dir | MOVX A, @DPTR | MOVX @DPTR, A
1 AJMP (P0) | ACALL (P0) | AJMP (P1) | ACALL (P1) | AJMP (P2) | ACALL (P2) | AJMP (P3) | ACALL (P3) | AJMP (P4) | ACALL (P4) | AJMP (P5) | ACALL (P5) | AJMP (P6) | ACALL (P6) | AJMP (P7) | ACALL (P7)
2 LJMP addr16 | LCALL addr16 | RET | RETI | ORL dir, A | ORL C, bit | ANL dir, A | XRL dir, A | JMP @A+DPTR | MOV C, bit | MOV bit, C | CLR bit | CLR bit | SETB bit | MOVX A, @R0 | MOVX @R0, A
3 RR A | RRC A | RL A | RLC A | ORL dir, # data | ANL dir, # data | XRL dir, # data | MOV dir, # data | MOVC A, @A+PC | INC DPTR | CPL C | CPL bit | SWAP A | DA A | SETB C | MOVX @R1, A
4 INC A | DEC A | ADD A, # data | ADDC A, # data | ORL A, # data | ANL A, # data | XRL A, # data | MOV A, # data | DIV AB | MUL AB | CJNE A, # data, rel | DJNZ dir, rel | CLR A | CPL A
5 INC dir | DEC dir | ADD A, dir | ADDC A, dir | ORL A, dir | ANL A, dir | XRL A, dir | MOV dir, # data | MOV dir, dir | SUBB A, dir | CJNE A, dir, rel | XCH A, dir | MOV A, dir | MOV dir, A
6 INC @R0 | DEC @R0 | ADD A, @R0 | ADDC A, @R0 | ORL A, @R0 | ANL A, @R0 | XRL A, @R0 | MOV @R0, # data | MOV dir, @R0 | SUBB A, @R0 | CJNE @R0, # data, rel | XCH A, @R0 | MOV A, @R0 | XCHD A, @R0 | MOV @R0, A
7 INC @R1 | DEC @R1 | ADD A, @R1 | ADDC A, @R1 | ORL A, @R1 | ANL A, @R1 | XRL A, @R1 | MOV @R1, # data | MOV dir, @R1 | SUBB A, @R1 | CJNE @R1, # data, rel | XCH A, @R1 | MOV A, @R1 | XCHD A, @R1 | MOV @R1, A
8 INC R0 | DEC R0 | ADD A, R0 | ADDC A, R0 | ORL A, R0 | ANL A, R0 | XRL A, R0 | MOV R0, # data | MOV dir, R0 | SUBB A, R0 | CJNE R0, # data, rel | XCH A, R0 | MOV A, R0 | DJNZ R0, rel | MOV R0, A
9 INC R1 | DEC R1 | ADD A, R1 | ADDC A, R1 | ORL A, R1 | ANL A, R1 | XRL A, R1 | MOV R1, # data | MOV dir, R1 | SUBB A, R1 | CJNE R1, # data, rel | XCH A, R1 | MOV A, R1 | DJNZ R1, rel | MOV R1, A
A INC R2 | DEC R2 | ADD A, R2 | ADDC A, R2 | ORL A, R2 | ANL A, R2 | XRL A, R2 | MOV R2, # data | MOV dir, R2 | SUBB A, R2 | CJNE R2, # data, rel | XCH A, R2 | MOV A, R2 | DJNZ R2, rel | MOV R2, A
B INC R3 | DEC R3 | ADD A, R3 | ADDC A, R3 | ORL A, R3 | ANL A, R3 | XRL A, R3 | MOV R3, # data | MOV dir, R3 | SUBB A, R3 | CJNE R3, # data, rel | XCH A, R3 | MOV A, R3 | DJNZ R3, rel | MOV R3, A
C INC R4 | DEC R4 | ADD A, R4 | ADDC A, R4 | ORL A, R4 | ANL A, R4 | XRL A, R4 | MOV R4, # data | MOV dir, R4 | SUBB A, R4 | CJNE R4, # data, rel | XCH A, R4 | MOV A, R4 | DJNZ R4, rel | MOV R4, A
D INC R5 | DEC R5 | ADD A, R5 | ADDC A, R5 | ORL A, R5 | ANL A, R5 | XRL A, R5 | MOV R5, # data | MOV dir, R5 | SUBB A, R5 | CJNE R5, # data, rel | XCH A, R5 | MOV A, R5 | DJNZ R5, rel | MOV R5, A
E INC R6 | DEC R6 | ADD A, R6 | ADDC A, R6 | ORL A, R6 | ANL A, R6 | XRL A, R6 | MOV R6, # data | MOV dir, R6 | SUBB A, R6 | CJNE R6, # data, rel | XCH A, R6 | MOV A, R6 | DJNZ R6, rel | MOV R6, A
F INC R7 | DEC R7 | ADD A, R7 | ADDC A, R7 | ORL A, R7 | ANL A, R7 | XRL A, R7 | MOV R7, # data | MOV dir, R7 | SUBB A, R7 | CJNE R7, # data, rel | XCH A, R7 | MOV A, R7 | DJNZ R7, rel | MOV R7, A
2Byte 3Byte
2Cycle 4Cycle

APPENDIX B: 8051 SPECIAL FUNCTION REGISTER (SFR)


Byte address
Bit address
FF
F0 F7 F6 F5 F4 F3 F2 F1 F0 B
E0 E7 E6 E5 E4 E3 E2 E1 E0 ACC
D0 D7 D6 D5 D4 D3 D2 D1 D0 PSW
B8 -- -- -- BC BB BA B9 B8 IP
B0 B7 B6 B5 B4 B3 B2 B1 B0 P3
A8 AF -- -- AC AB AA A9 A8 IE
A0 A7 A6 A5 A4 A3 A2 A1 A0 P2
99 not bit addressable SBUF
98 9F 9E 9D 9C 9B 9A 99 98 SCON
90 97 96 95 94 93 92 91 90 P1
8D not bit addressable TH1
8C not bit addressable TH0
8B not bit addressable TL1
8A not bit addressable TL0
89 not bit addressable TMOD
88 8F 8E 8D 8C 8B 8A 89 88 TCON
87 not bit addressable PCON
83 not bit addressable DPH
82 not bit addressable DPL
81 not bit addressable SP
80 87 86 85 84 83 82 81 80 P0
Special Function Registers

SFR RAM Address (Byte and Bit)

APPENDIX C: BIT-ADDRESSABLE RAM LOCATIONS

Byte Address	Bit Address							
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00

APPENDIX D: PROGRAM STATUS WORD (PSW)

CY	AC	F0	RS1	RS0	OV	--	P

APPENDIX E: TIMER/COUNTER MODE CONTROL (TMOD) REGISTER

G	C/T	M1	M0	G	C/T	M1	M0

APPENDIX F: SERIAL CONTROL (SCON) REGISTER

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

APPENDIX G: INTERRUPT ENABLE (IE) REGISTER

EA	--	ET2	ES	ET1	EX1	ET0	EX0

APPENDIX H: INTERRUPT PRIORITY (IP) REGISTER

--	--	PT2	PS	PT1	PX1	PT0	PX0